



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/870,621	05/31/2001	Scott J. Broussard	AUS920010263US1	1785

35617 7590 06/30/2004

CONLEY ROSE, P.C.
P.O. BOX 684908
AUSTIN, TX 78768

EXAMINER

BONSHOCK, DENNIS G

ART UNIT	PAPER NUMBER
----------	--------------

2173

DATE MAILED: 06/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/870,621

Applicant(s)

BROUSSARD, SCOTT J.

Examiner

Dennis G. Bonshock

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 April 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-25 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____

Final Rejection

1. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment A as received on 04-15-2004.

2. Claims 1-25 have been examined.

Status of Claims:

3. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over WinZip computing Inc., WINZIP 8.0, hereinafter WinZip, Java, and Fults et al. Patent #5327529, hereinafter Fults.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over WinZip computing Inc., WINZIP 8.0, hereinafter WinZip, Java, and Fults et al. Patent #5327529, hereinafter Fults.

6. With regard to claim 1, WinZip teaches, on pages 3 and 4 a system of software components adapted to display text running under an operating system, in which selection of the mask password check box displays the text with one or two software components namely masked ("*****") or unmasked ("password"), the selection of which is made at runtime. Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the

Art Unit: 2173

JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). WinZip and Java, however don't teach using a peer component for selection between two proxy components. Fults teaches a system which generates the interface based on a selection by the user (see column 3, lines 27-52) similar to that of WinZip and Java, Fults, however, further teaches taking hints from a user and using the hints to direct the interface generation to an appropriate user interface implementation (see column 3, lines 33-51, figure 2, and the abstract). Fults further teaches, in column 20, lines 24-34, that libraries are dynamically loaded into memory when needed by the application. It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces. It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Java, and Fults, before him at the time the invention was made to modify the password display system of WinZip and Java to be implemented with a peer component that directs the interface development to the appropriate user interface implementation. One would have been motivated to make such a combination because WinZip has a switch of display characteristics very similar to that of Fults, where upon a user selection, an element is displayed using a different API component.

Art Unit: 2173

7. With regard to claims 2 and 14, which teach the object is part of a graphical user interface associated with the application program, WinZip teaches, on pages 3 and 4, that the object is part of a GUI associated with the application program WinZip.

8. With regard to claims 3 and 15, which teach the selection of the first or second of the systems of software components being made by the application, by sending an indication of the mode of use to the peer component, Fults teaches, in column 3, lines 33-51, figure 2, and the abstract, the indication of the specific user interface element being made to the peer component (the Generic User Interface Design).

9. With regard to claim 4 and 16, which teaches that the new graphics resource component is created during runtime, upon selection of the first or second proxy components to replace a graphics resource component previously created for the display object, Fults further teaches, in column 20, lines 24-34, that libraries are dynamically loaded into memory when needed by the application.

10. With regard to claims 5 and 17, which teach an instance of the first or second of the system of software components previously selected by the application program is destroyed when the instance is no longer selected, WinZip teaches, on pages 3 and 4, that selection of the software used to display the text is made in the application program WinZip through the use of the checkbox, and if there is existing text in the box when selection is made it will be destroyed and the new selected format will be displayed.

Art Unit: 2173

11. With regard to claims 6 and 18, which teach the appearance and behavior of the object differ depending on whether the object is displayed by the first or second of the system of software components (proxy components), WinZip further teaches, on page 3 and 4, that the appearance is different whether or not the mask is checked. WinZip also teaches, on page 6, that if the password is masked the user will be required to type it twice.

12. With regard to claims 7 and 19, which teach that the application program is written in Java programming language, WinZip and Fults teaches the system for displaying the text in two different formats, but doesn't teach the use of Java programming language. Java teaches a system of masking password similar to that of WinZip, but further teaches the use of Java programming language, specifically the Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Fults, and Java before him at the time the invention was made to modify the text display system of WinZip and Fults to use Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

13. With regard to claims 8 and 20, which teach the graphic resource components are included in a set of software components comprising the Swing application program interface (API), WinZip teaches the system for displaying the text in two different formats, but doesn't teach the use of Swing API. Java teaches a system of masking passwords similar to that of WinZip but further

Art Unit: 2173

teaches the use of Java's Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to use Java's Swing API. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

14. With regard to claims 9 and 22, which teach the operating system comprises Windows, Unix or OS/2 computer operating system, WinZip and Fults teaches the system for displaying the text in two different formats, but doesn't teach the operating system comprising Windows, Unix or OS/2 computer operating system. Java teaches a system of masking password similar to that of WinZip, but teaches the use of Java's Swing API which can be displayed with the same look and feel on Windows, Unix or Apple computers through the use of a components set code-named *Metal* (see *IS page 1, paragraphs 1- 5*). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Fults, and Java before him at the time the invention was made to modify the text display system of WinZip to use Java's Swing API for platform independence. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

Art Unit: 2173

15. With regard to claims 10 and 21, which teach the first proxy component using JTextField and the second proxy component using the JPasswordField, of the Swing application program interface (API), WinZip and Fults teach the system for displaying the text in two different formats, but doesn't teach the use of a TextField and PasswordField. Java teaches a system of masking passwords similar to that of WinZip but teaches the use of Java's Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Fults and Java before him at the time the invention was made to modify the text display system of WinZip and Fults to use Java's Swing API. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

16. With regard to claims 11 and 23, which teach the selection of either the first or second proxy components depends on the status of a software flag associated with the object, WinZip further teaches on page 3 and 4, that the appearance is different whether or not the mask is checked. Java Platform 1.2 Beta 4 API Specification: Class JPasswordField also teaches this in paragraph 1.

17. With regard to claims 12 and 24, which teach that the object is adapted to respond to text entry events and wherein the status of the software flag indicates whether or not a special character is echoed when text is entered, WinZip further teaches on page 3 and 4, that the appearance is different whether or not the mask is checked (characters are displayed if unselected, asterisks are displayed

Art Unit: 2173

if selected). Java Platform 1.2 Beta 4 API Specification: Class JPasswordField also teaches this in paragraph 1 specifically mentioning an echoChar.

18. With regard to claim 13, WinZip teaches, on pages 3 and 4 a system of software components adapted to display text running under an operating system, in which selection of the mask password check box displays the text with one or two software components namely masked ("*****") or unmasked ("password"), the selection of which is made at runtime. WinZip further teaches typing in a character set with the first software component, monitoring the mask flag, and then changing the mode of use of the object upon selection of the flag with an appearance distinct from the first (see pages 3 and 4), and also that if the mask is selected the password must be typed twice (see page 6). Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). WinZip and Java, however don't teach using a peer component for selection between two proxy components. Fults teaches a system which generates the interface based on a selection by the user (see column 3, lines 27-52) similar to that of WinZip and Java, Fults, however, further teaches taking hints from a user and using the hints to direct the interface generation to an appropriate user interface implementation (see column 3, lines 33-51, figure 2, and the abstract). Fults further teaches, in column 20, lines 24-34, that libraries are dynamically loaded into memory when needed by the application. It would have been obvious to one

Art Unit: 2173

of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces. It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Java, and Fults, before him at the time the invention was made to modify the password display system of WinZip and Java to be implemented with a peer component that directs the interface development to the appropriate user interface implementation. One would have been motivated to make such a combination because WinZip has a switch of display characteristics very similar to that of Fults, where upon a user selection, an element is displayed using a different API component.

19. With regard to claim 25, WinZip teaches, on pages 3 and 4 a windows based operating system with a system of software components adapted to display text in an application program, in which selection of the mask password check box displays the text with one or two software components namely masked ("*****") or unmasked ("password"), the selection of which is made at runtime. WinZip further teaches typing in a character set with the first software component, monitoring the mask flag, and then changing the mode of use of the object upon selection of the flag with an appearance distinct from the first (see pages 3 and 4), and also that if the mask is selected the password must be typed twice (see page 6). Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the JPasswordField and JTextField

Art Unit: 2173

(see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). WinZip and Java, however don't teach using a peer component for selection between two proxy components. Fults teaches a system which generates the interface based on a selection by the user (see column 3, lines 27-52) similar to that of WinZip and Java, Fults, however, further teaches taking hints from a user and using the hints to direct the interface generation to an appropriate user interface implementation (see column 3, lines 33-51, figure 2, and the abstract). Fults further teaches, in column 20, lines 24-34, that libraries are dynamically loaded into memory when needed by the application. It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces. It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip, Java, and Fults, before him at the time the invention was made to modify the password display system of WinZip and Java to be implemented with a peer component that directs the interface development to the appropriate user interface implementation. One would have been motivated to make such a combination because WinZip has a switch of display characteristics very similar to that of Fults, where upon a user selection, an element is displayed using a different API component.

Response to Arguments

20. The arguments filed on 04-15-2004 have been fully considered but they are not persuasive. Reasons set forth below.

21. The applicants' argue that with respect to claims 1, 13, and 25, that "none of the cited art teach or suggest a system of software components that include a first proxy component, a second proxy component and a peer component for displaying an object where the peer component is configured for selection, during runtime, either the first proxy component or the second proxy component, depending on a mode of use of the object, and where the selected proxy component dynamically creates a new graphics resource component for displaying the object such that the appearance of the object is substantially independent of an operating system. "

22. Applicant's arguments with respect to claims 1, 13, and 25 have been considered but are moot in view of the new ground(s) of rejection.

23. The applicants' argue that "the appropriate Swing component must be dynamically recreated and substituted for the AWT component each time the mode of user of the object changes."

24. Applicant's arguments with respect to the above-cited limitation have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

25. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.**

Art Unit: 2173

See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

26. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

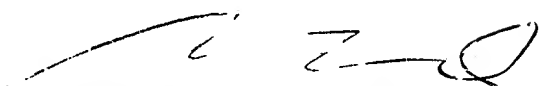
27. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G. Bonshock whose telephone number is (703) 305-4668. The examiner can normally be reached on Monday - Friday, 6:30 a.m. - 4:00 p.m.

28. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2173

29. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

dgb



RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173